CLASE 5 : NVIC

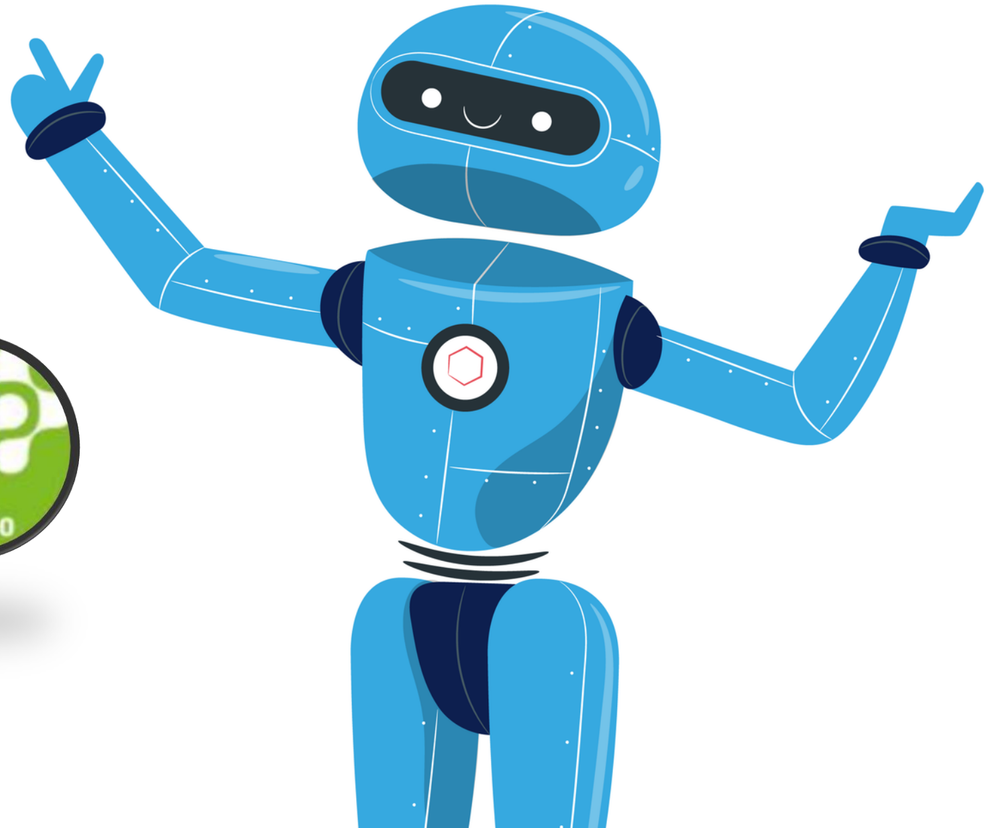
MICROCONTROLADORES

ARM

life.augmented

STM32 L4

STM32 Nucleo



TEXAS INSTRUMENTS

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

DESCRIPCION DEL NVIC

arm

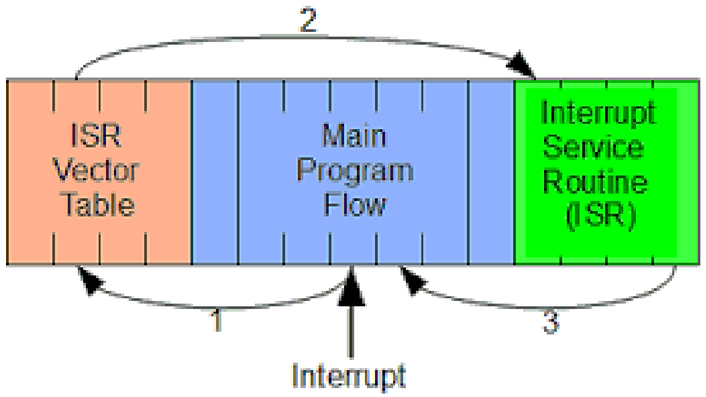
MICRO-
CONTRO-
LADORES
ARM

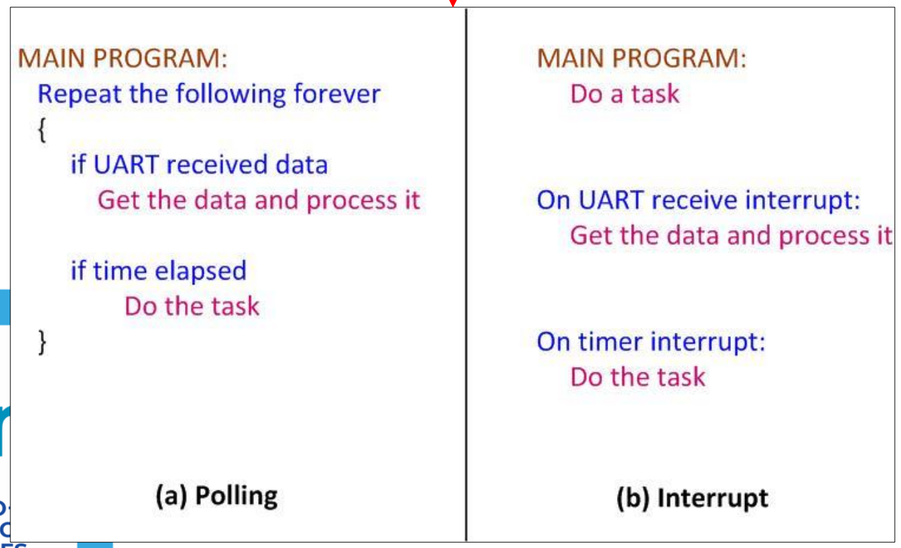
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

Interrumpe el flujo normal del programa principal.





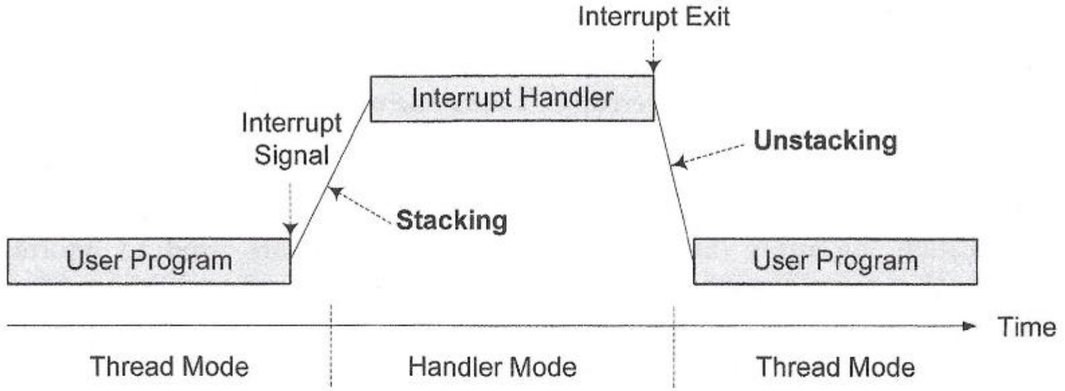
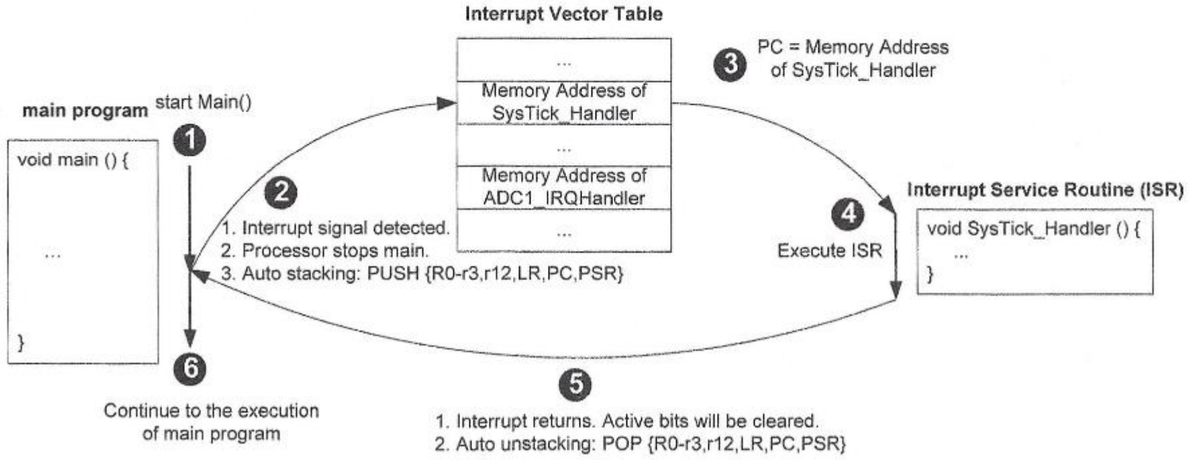


Figure 11-3. Automatic stacking and unstacking for interrupt handler



CONTROL Register in ARM Cortex-M4

nPRIV (Privilege): Defines the Thread mode privilege level

0: Privileged

1: Unprivileged

Active Stack Pointer (ASP): Defines the currently active stack pointer (ASP = SPSEL)

0: MSP is the current stack pointer.

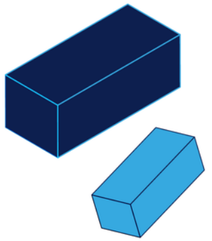
1: PSP is the current stack pointer.

Floating Point Context Active (FPCA)

0: No floating point context active.

1: Floating point context active.

Privileged level Execution and Processor Modes in ARM Cortex-M

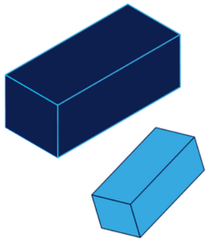


Processor Mode	Software	Privilege level
Thread	Applications	Privileged and Unprivileged
Handler	ISR for Exceptions and IRQs	Always Privileged
In Thread mode, use bit 0 of the CONTROL register to select Privileged or Unprivileged		

arm
MICRO-CONTRO-LADORES ARM

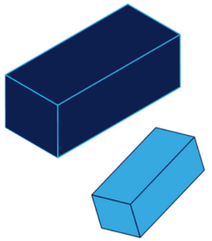
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Processor Modes and Stack Usage in ARM Cortex-M



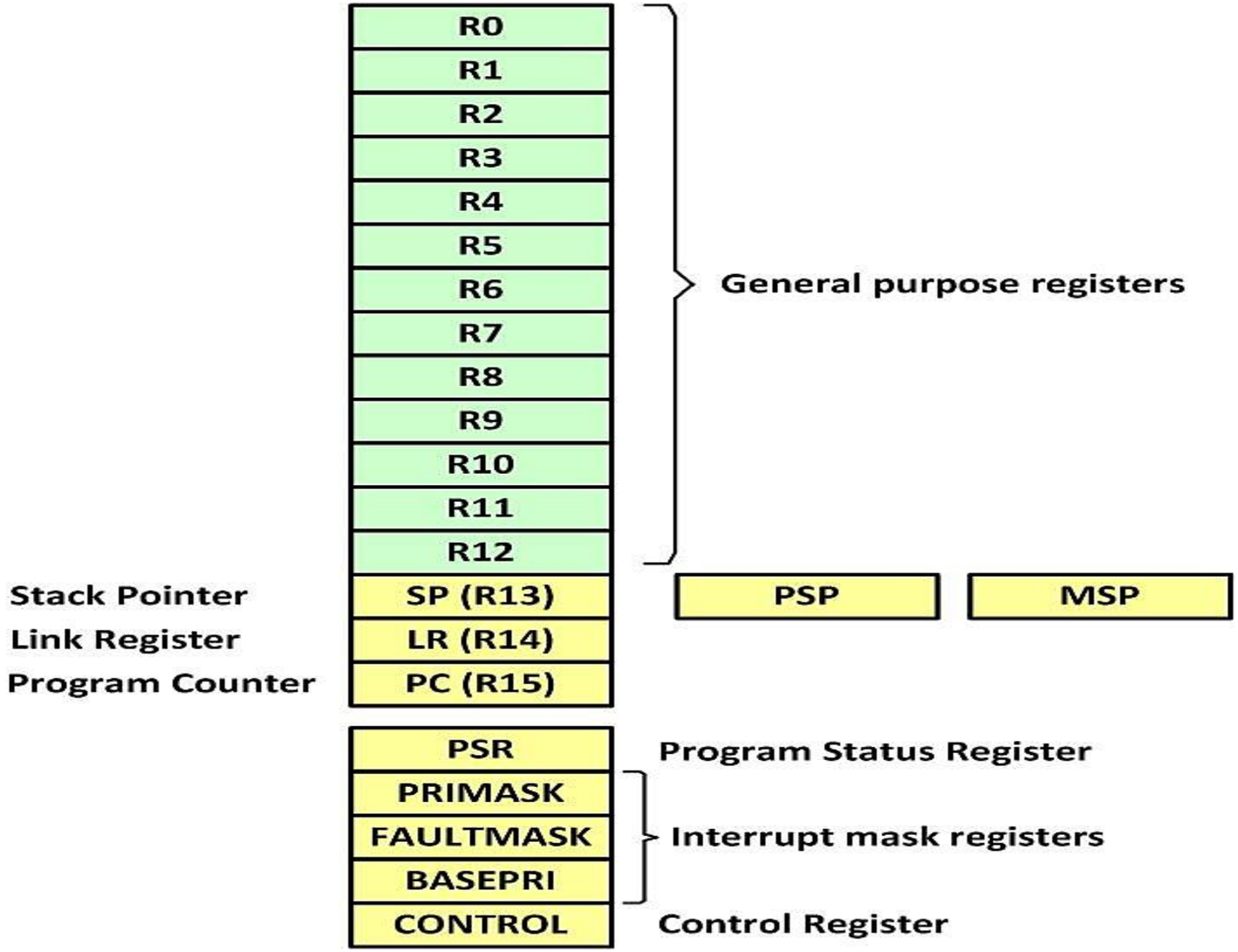
Processor Mode	Software	Stack Usage
Thread	Applications	MSP or PSP
Handler	ISR for Exceptions and IRQs	MSP
Note: In Thread mode, use bit 1 of the Control register to select MSP or PSP for stack pointer.		

Processor Mode, Privilege, and Stack in ARM Cortex



Mode	Privilege	Stack Pointer	Typical Example usage
Handler	Privileged	Main	Exception Handling
Handler	Unprivileged	Any	Reserved since Handler is always Privileged
Thread	Privileged	Main	Operating system kernel
Thread	Privileged	Process	
Thread	Unprivileged	Main	
Thread	Unprivileged	Process	Application threads

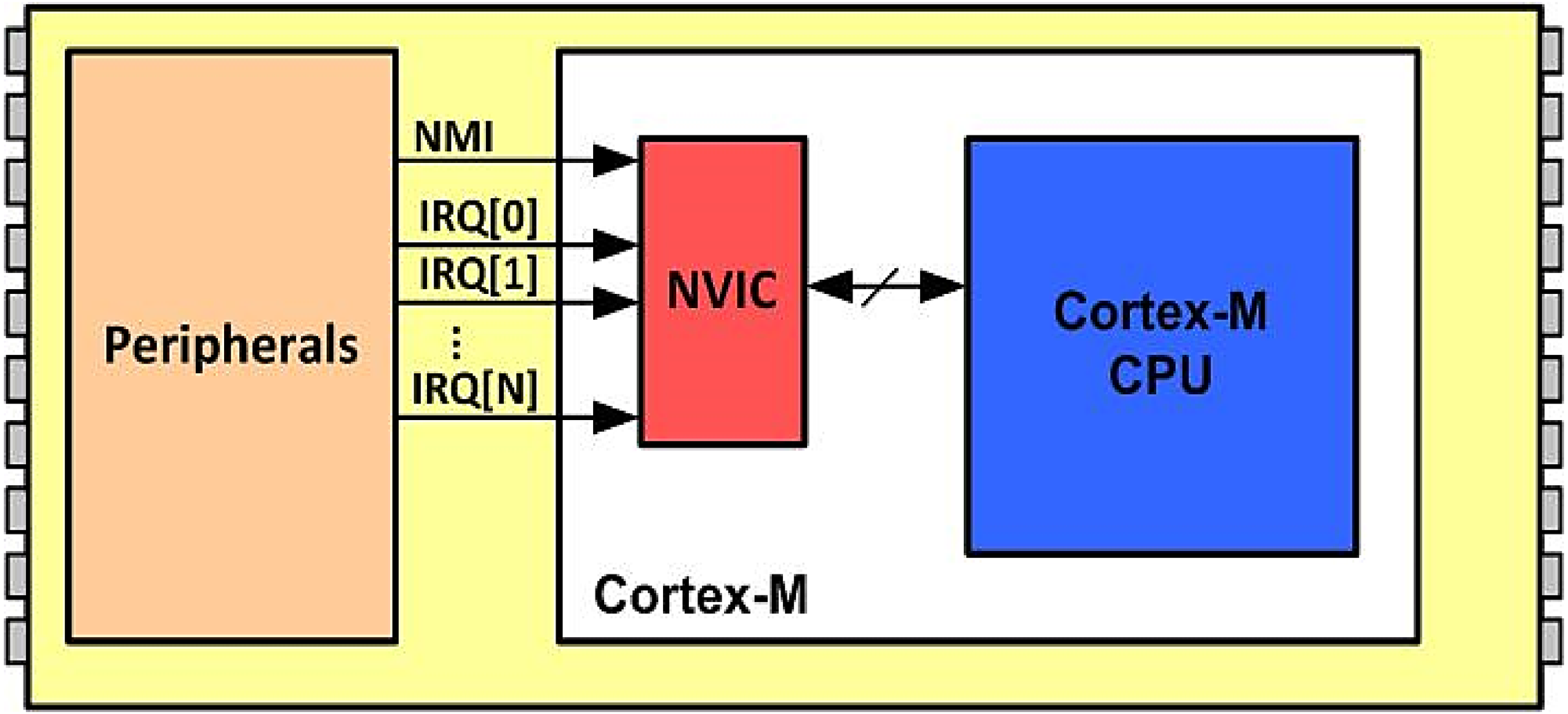
ARM Cortex-M Registers



Special function registers of ARM Cortex-M

Register name	Privilege Usage
MSP (main stack pointer)	Privileged
PSP (processor stack pointer)	Privileged or Unprivileged
PSR (Processor status register)	Privileged
APSR (application processor status register)	Privileged or Unprivileged
ISPR (interrupt processor status register)	Privileged
EPSR (execution processor status register)	Privileged
PRIMASK (Priority Mask register)	Privileged
FAULTMASK(fault mask register)	Privileged
BASEPRI (base priority register)	Privileged
CONTROL (control register)	Privileged
Note: We must use MSR and MRS instructions to access the above registers	

INTERRUPCIONES CORTEX-M



arm
MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **NVIC**

La NVIC es un periférico interno del CORE ARM CORTEX-M y es el encargado de manejar todas las interrupciones. Realiza estas tres funciones:

1. Habilita y deshabilita las interrupciones.
2. Configura la prioridad y sub-prioridad de una determinada interrupción.
3. Establece y limpia el handling bit de una interrupción.

Interrupt control bit	Corresponding register (32 bits)
Enable bit	Interrupt set enable register (ISER)
Disable bit	Interrupt clear enable register (ICER)
Pending bit	Interrupt set pending register (ISPR)
Un-pending bit	Interrupt clear pending register (ICPR)
Active bit	Interrupt active bit register (IABR)
Software trigger bit	Software trigger interrupt register (STIR)

INTERRUPCIONES CORTEX-M

ESTADOS DE EXEPCIONES

- **Inactivo:** La excepción no está activa ni pendiente.
- **Pendiente:** La excepción está a la espera de que la procese el procesador.
- **Activo:** Una excepción que está siendo atendida por el procesador pero que no se ha completado.
- **Activo y Pendiente:** la excepción está siendo atendida por el procesador y hay un pendiente de excepción de la misma fuente.

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **NUMERO DE INTERRUPCION (IRQn)**

Los procesadores ARM CORTEX-M soportan hasta 240 tipos de interrupciones, excluyendo la interrupción del reset, son identificados con un único número en el rango de -15 a 240. El numero de interrupciones son definidos por el fabricante del MCU ARM.

```
  Reset_IRQn                = -15,              /*!<   1  Reset Vector, invoked on Power up and warm reset                */
  NonMaskableInt_IRQn       = -14,              /*!<   2  Non maskable Interrupt, cannot be stopped or preempted          */
  HardFault_IRQn            = -13,              /*!<   3  Hard Fault, all classes of Fault                                */
  MemoryManagement_IRQn     = -12,              /*!<   4  Memory Management, MPU mismatch, including Access Violation
                                                    and No Match                                                            */
  BusFault_IRQn             = -11,              /*!<   5  Bus Fault, Pre-Fetch-, Memory Access Fault, other address/memory
                                                    related Fault                                                           */
  UsageFault_IRQn           = -10,              /*!<   6  Usage Fault, i.e. Undef Instruction, Illegal State Transition   */
  SVCall_IRQn               =  -5,              /*!<  11  System Service Call via SVC instruction                         */
  DebugMonitor_IRQn         =  -4,              /*!<  12  Debug Monitor                                                   */
  PendSV_IRQn               =  -2,              /*!<  14  Pendable request for system service                             */
  SysTick_IRQn              =  -1,              /*!<  15  System Tick Timer                                               */
/* ------------------  TM4C123GH6PM Specific Interrupt Numbers  ------------------ */
  GPIOA_IRQn                =   0,              /*!<   0  GPIOA                                                           */
  GPIOB_IRQn                =   1,              /*!<   1  GPIOB                                                           */
  GPIOC_IRQn                =   2,              /*!<   2  GPIOC                                                           */
  GPIOD_IRQn                =   3,              /*!<   3  GPIOD                                                           */
  GPIOE_IRQn                =   4,              /*!<   4  GPIOE                                                           */
  UART0_IRQn                =   5,              /*!<   5  UART0                                                           */
  UART1_IRQn                =   6,              /*!<   6  UART1                                                           */
```

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- **RUTINAS DE SERVICIO DE INTERRUPCION (ISR)**

Son rutinas especiales que son invocados automáticamente en respuesta a una interrupción. Cada rutina de servicio de interrupción esta definido por defecto en el system startup. (startup_stm32f401xe.s).

```
SysTick_Handler PROC
                EXPORT  SysTick_Handler            [WEAK]
                B       .
                ENDP
```

MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

TABLA DE INTERRUPCIONES

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- TABLA DE VECTOR DE INTERRUPCIONES

DEFINIDO POR EL FABRICANTE

Exception Type	ARMv6-M (Cortex-M0/M0+/M1)	ARMv7-M (Cortex-M3/M4/M7)	ARMv8-M Baseline (Cortex-M23)	ARMv8-M Mainline (Cortex-M33)	Vector Table	Vector address offset (initial)
495		Not supported in Cortex-M3/M4/M7	Not supported in Cortex-M23	Device Specific Interrupts	Interrupt#479 vector [1]	0x000007BC
256						
255		Device Specific Interrupts	Device Specific Interrupts		Interrupt#239 vector [1]	0x000003FC
31	Device Specific Interrupts				Interrupt#31 vector [1]	0x000000BC
17					Interrupt#1 vector [1]	0x00000044
16					Interrupt#0 vector [1]	0x00000040
15	SysTick	SysTick	SysTick	SysTick	SysTick vector [1]	0x0000003C
14	PendSV	PendSV	PendSV	PendSV	PendSV vector [1]	0x00000038
13	Not used	Not used	Not used	Not used	Not used	0x00000034
12		Debug Monitor		Debug Monitor	Debug Monitor vector [1]	0x00000030
11	SVC	SVC	SVC	SVC	SVC vector [1]	0x0000002C
10	Not used	Not used	Not used	Not used	Not used	0x00000028
9					Not used	0x00000024
8					Not used	0x00000020
7				SecureFault	SecureFault (ARMv8-M Mainline) [1]	0x0000001C
6		Usage Fault		Usage Fault	Usage Fault vector [1]	0x00000018
5		Bus Fault		Bus Fault	Bus Fault vector [1]	0x00000014
4		MemManage (fault)		MemManage (fault)	MemManage vector [1]	0x00000010
3	HardFault	HardFault	HardFault	HardFault	HardFault vector [1]	0x0000000C
2	NMI	NMI	NMI	NMI	NMI vector [1]	0x00000008
1					Reset vector [1]	0x00000004
0					MSP initial value	0x00000000

arm MICRO-CONTRO-LADORES ARM

ER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- TABLA DE VECTOR DE INTERRUPCIONES

DEFINIDO COMO: -15 al -1 en la librería CMSIS

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description
0-15	-	0x0000.0000 - 0x0000.003C	Processor exceptions
16	0	0x0000.0040	GPIO Port A
17	1	0x0000.0044	GPIO Port B
18	2	0x0000.0048	GPIO Port C
19	3	0x0000.004C	GPIO Port D
20	4	0x0000.0050	GPIO Port E
21	5	0x0000.0054	UART0
22	6	0x0000.0058	UART1
23	7	0x0000.005C	SSI0
24	8	0x0000.0060	I^2C0
25	9	0x0000.0064	PWM0 Fault
26	10	0x0000.0068	PWM0 Generator 0
27	11	0x0000.006C	PWM0 Generator 1
28	12	0x0000.0070	PWM0 Generator 2
29	13	0x0000.0074	QEI0
30	14	0x0000.0078	ADC0 Sequence 0
31	15	0x0000.007C	ADC0 Sequence 1
32	16	0x0000.0080	ADC0 Sequence 2

DEFINIDO POR LA ARQUITECTURA ARM CORTEX-M

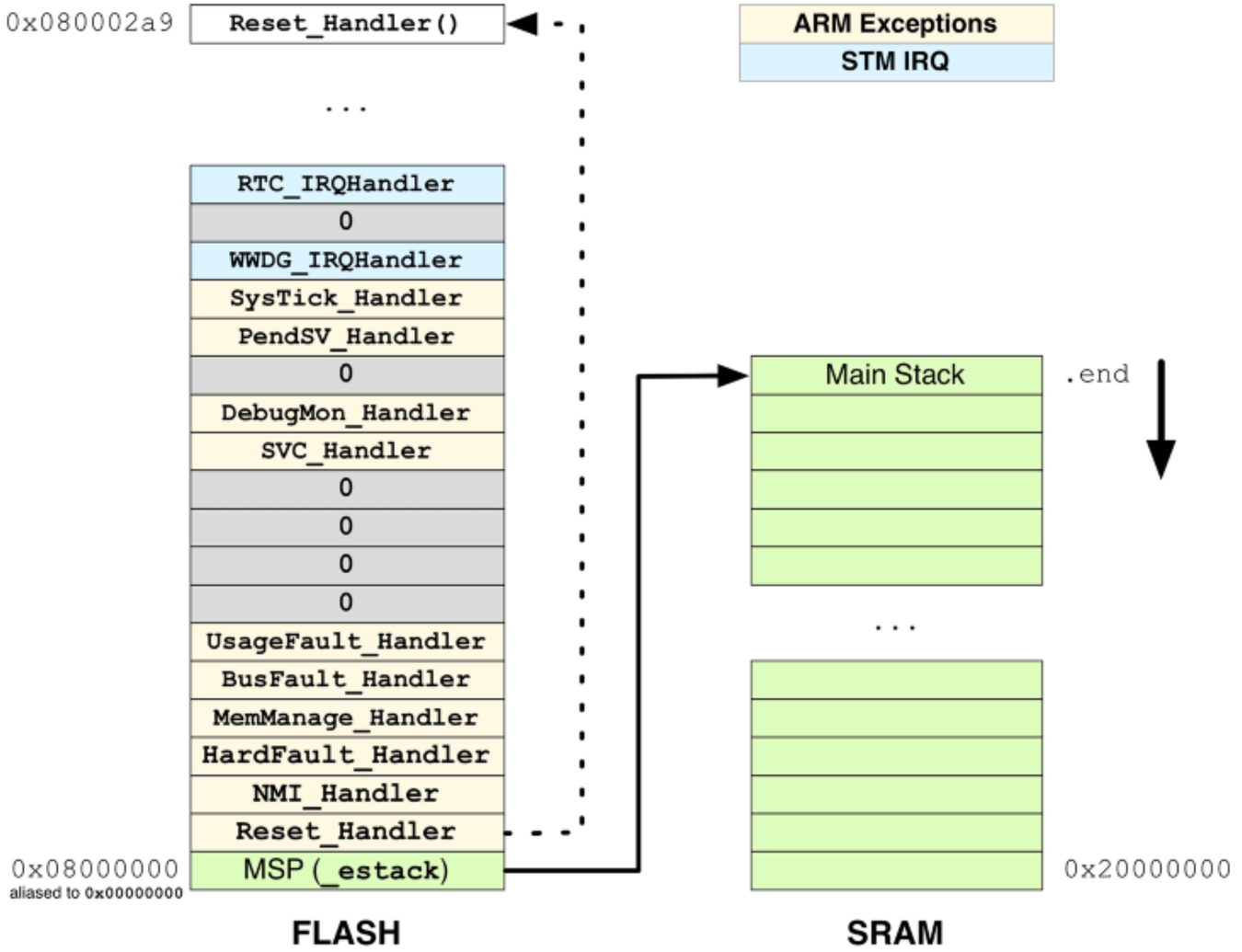
arm MICRO-CONTRO-LADORES ARM

ER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

INTERRUPCIONES

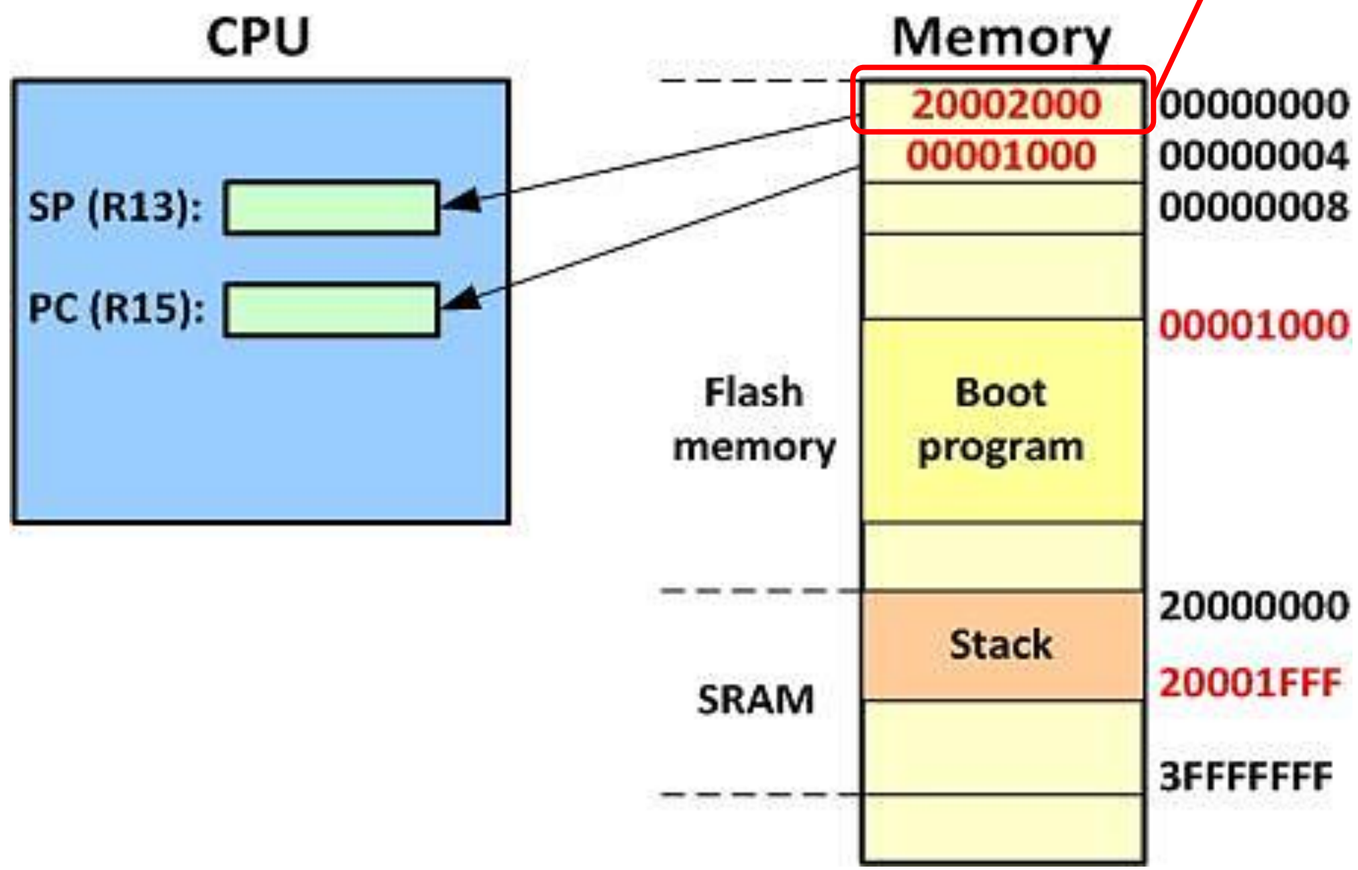
- TABLA DE VECTOR DE INTERRUPCIONES



INTERRUPCIONES CORTEX-M

INTERRUPCIONES

- TABLA DE VECTOR DE INTERRUPCIONES



PRIORIDAD Y SUBPRIORIDAD

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

PRIORIDAD

PRIORIDAD ALTA

PRIORIDAD BAJA

IRQn	Interrupt	Priority Level
0	Stack Pointer initial value	
1	Reset	-3 Highest
2	NMI	-2
3	Hard Fault	-1
4	Memory Management Fault	Programmable
5	Bus Fault	Programmable
6	Usage Fault (undefined instructions, divide by zero, unaligned memory access,....)	Programmable
7	Reserved	Programmable
8	Reserved	Programmable
9	Reserved	Programmable
10	Reserved	Programmable
11	SVCall	Programmable
12	Debug Monitor	Programmable
13	Reserved	Programmable
14	PendSV	Programmable
15	SysTick	Programmable
16	IRQ 0 for peripherals	Programmable
17	IRQ 1 for peripherals	Programmable
…	…	Programmable
255	IRQ 239 for peripherals	Programmable

arm MICRO-CONTRO-LADORES ARM

INTERRUPCIONES CORTEX-M

PRIORIDAD

Configurable priority values are in the range 0-15.

- System handler priority registers (SHPRx)

Handler	Field	Register description
Memory management fault	PRI_4	*System handler priority register 1 (SHPR1)*
Bus fault	PRI_5	
Usage fault	PRI_6	
SVCall	PRI_11	*System handler priority register 2 (SHPR2) on page 233*
PendSV	PRI_14	*System handler priority register 3 (SHPR3) on page 234*
SysTick	PRI_15	

- Interrupt priority register x (NVIC_IPRx)

INTERRUPCIONES CORTEX-M

PRIORIDAD Y SUBPRIORIDAD

Para aumentar el control de prioridad en sistemas con interrupciones, el NVIC admite la agrupación de prioridades.

- Un campo superior que define la prioridad del grupo.
- Un campo inferior que define una subprioridad dentro del grupo.

Application interrupt and reset control register (APINT)

Address offset: 0x0C
Reset value: 0xFA05 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
r					rw	rw	rw						w	w	w

Bits 10:8 **PRIGROUP**: Interrupt priority grouping field
This field determines the split of group priority from subpriority, see *Binary point on page 228*.

PRIGROUP Bit Field	Binary Point[a]	Group Priority Field	Subpriority Field	Group Priorities	Subpriorities
0x0 - 0x4	bxxx.	[7:5]	None	8	1
0x5	bxx.y	[7:6]	[5]	4	2
0x6	bx.yy	[7]	[6:5]	2	4
0x7	b.yyy	None	[7:5]	1	8

REGITROS PARA PROGRAMAR LA NVIC

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Address	Name	Type	Required privilege	Reset value	Description
0xE000E100-0xE000E11F	NVIC_ISER0-NVIC_ISER7	RW	Privileged	0x00000000	*Table 4.3.2: Interrupt set-enable register x (NVIC_ISERx) on page 210*
0XE000E180-0xE000E19F	NVIC_ICER0-NVIC_ICER7	RW	Privileged	0x00000000	*Table 4.3.3: Interrupt clear-enable register x (NVIC_ICERx) on page 211*
0XE000E200-0xE000E21F	NVIC_ISPR0-NVIC_ISPR7	RW	Privileged	0x00000000	*Table 4.3.4: Interrupt set-pending register x (NVIC_ISPRx) on page 212*
0XE000E280-0xE000E29F	NVIC_ICPR0-NVIC_ICPR7	RW	Privileged	0x00000000	*Table 4.3.5: Interrupt clear-pending register x (NVIC_ICPRx) on page 213*
0xE000E300-0xE000E31F	NVIC_IABR0-NVIC_IABR7	RW	Privileged	0x00000000	*Table 4.3.6: Interrupt active bit register x (NVIC_IABRx) on page 214*
0xE000E400-0xE000E4EF	NVIC_IPR0-NVIC_IPR59	RW	Privileged	0x00000000	*Table 4.3.7: Interrupt priority register x (NVIC_IPRx) on page 215*
0xE000EF00	STIR	WO	Configurable	0x00000000	*Table 4.3.8: Software trigger interrupt register (NVIC_STIR) on page 216*

PDF: Programming manual STM32L4

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt set-enable register x (NVIC_ISERx)

```
Address offset: 0x100 + 0x04 * x, (x = 0 to 7)
Reset value: 0x0000 0000
Required privilege: Privileged
NVIC_ISER0 bits 0 to 31 are for interrupt 0 to 31, respectively
NVIC_ISER1 bits 0 to 31 are for interrupt 32 to 63, respectively
....
NVIC_ISER6 bits 0 to 31 are for interrupt 192 to 223, respectively
NVIC_ISER7 bits 0 to 15 are for interrupt 224 to 239, respectively
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETENA[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETENA[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

Bits 31:0 **SETENA**: Interrupt set-enable bits.

Write:

0: No effect

1: Enable interrupt

Read:

0: Interrupt disabled

1: Interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

Bits 16 to 31 of the NVIC_ISER7 register are reserved.

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt clear-enable register x (NVIC_ICERx))

```
Address offset: 0x180 + 0x04 * x, (x = 0 to 7)
Reset value: 0x0000 0000
Required privilege: Privileged
NVIC_ICER0 bits 0 to 31 are for interrupt 0 to 31, respectively
NVIC_ICER1 bits 0 to 31 are for interrupt 32 to 63, respectively
....
NVIC_ICER6 bits 0 to 31 are for interrupt 192 to 223, respectively
NVIC_ICER7 bits 0 to 15 are for interrupt 224 to 239, respectively
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits 31:0 **CLRENA**: Interrupt clear-enable bits.

Write:

0: No effect

1: Disable interrupt

Read:

0: Interrupt disabled

1: Interrupt enabled.

Bits 16 to 31 of the NVIC_ICER7 register are reserved.

INTERRUPCIONES CORTEX-M

REGISTROS DE LA NVIC

Interrupt priority register x (NVIC_IPRx)

Address offset: 0x400 + 0x04 * x, (x = 0 to 59)
Reset value: 0x0000 0000
Required privilege: Privileged

	31 – 24	23 – 16	15 – 8	7 – 0
NVIC_IPR59	IP[239]	IP[238]	IP[237]	IP[236]
NVIC_IPRx	IP[4x+3]	IP[4x+2]	IP[4x+1]	IP[4x]
NVIC_IPR0	IP[3]	IP[2]	IP[1]	IP[0]

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

- **HABILITAR/DESHABILITAR INTERRUPCIONES DE PERIFERICOS**

Escribir un 1 en el registro **ISER** habilita una interrupción y escribir un 1 en el registro **ICER.**

$$byteOffset = floor\left(\frac{\text{Numero de Interrupción (IRQn)}}{32}\right)$$

$$byteOffset = \text{Numero de Interrupción (IRQn)} >> 5U$$

$$BitOffset = \text{Numero de Interrupción (IRQn)} \ \& \ 0x1F$$

Habilitar la interrupcion para el numero de interrupcion 77:

$$BitOffset = floor\left(\frac{72}{32}\right) = 2$$

$$BitOffset = 77 \ \& \ 31$$

$$BitOffset = 13$$

$$NVIC->ISER[byteOffset] = 1 << bitOffset;$$

$$NVIC->ISER[2] = 1 << 13;$$

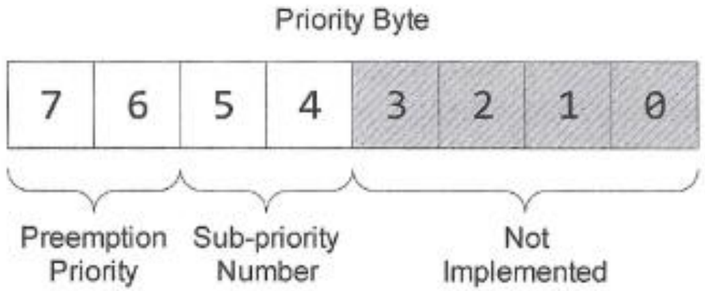
en general:

$$NVIC->ISER[IRQn >> 5U] = 1U << (XXX\_IRQn \ \& \ 0x1F);$$

INTERRUPCIONES CORTEX-M

- **ESTABLECER LA PRIORIDAD DE LAS INTERRUPCIONES DE LOS PERIFERICOS**

Cada interrupción cuenta con su registro de interrupción (IP), El cual tiene mapeado 8bits por interrupción. **Un valor bajo de un numero de prioridad representa una prioridad.**



$$NVIC->IP[XXXX\_IRQn] = priority << (4\ \&\ 0xFF);$$

prioridad de exepciones del sistema

```
SCB->SHP[(((uint32_t)IRQn) & 0xFUL)-4UL] = (uint8_t)((priority << (8U - __NVIC_PRIO_BITS)) & (uint32_t)0xFFUL);
SCB->SHP[(((uint32_t)IRQn) & 0xFUL)-4UL] = (uint8_t)((priority << (4)) & (uint32_t)0xFFUL);
```

arm
MICRO-CONTRO-LADORES ARM

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

INTERRUPCIONES CORTEX-M

FUNCIONES CMSIS PARA EL MANEJO DE INTERRUPCIONES

El software usa las instrucciones **CPSEI** y **CPSID** para habilitar y deshabilitar las interrupciones.

```
void __disable_irq(void) // Disable Interrupts
void __enable_irq(void) // Enable Interrupts
```

En adición la CMSIS incluye las funciones:

CMSIS interrupt control function	Description
void NVIC_SetPriorityGrouping(uint32_t priority_grouping)	Set the priority grouping
void NVIC_EnableIRQ(IRQn_t IRQn)	Enable IRQn
void NVIC_DisableIRQ(IRQn_t IRQn)	Disable IRQn
uint32_t NVIC_GetPendingIRQ (IRQn_t IRQn)	Return true (IRQ-Number) if IRQn is pending
void NVIC_SetPendingIRQ (IRQn_t IRQn)	Set IRQn pending
void NVIC_ClearPendingIRQ (IRQn_t IRQn)	Clear IRQn pending status
uint32_t NVIC_GetActive (IRQn_t IRQn)	Return the IRQ number of the active interrupt
void NVIC_SetPriority (IRQn_t IRQn, uint32_t priority)	Set priority for IRQn
uint32_t NVIC_GetPriority (IRQn_t IRQn)	Read priority of IRQn
void NVIC_SystemReset (void)	Reset the system

TivaWare

COMUNICACIÓN SERIAL - UART

INTERRUPCIONES

- **REGISTRO DE PRIORIDAD Y HABILITACION DE INTERRUPCION (NVIC)**
  - **PRIORIDAD**

UART0 → NVIC_PRI1_R [15:13]
UART1 → NVIC_PRI1_R [23:21]
UART2 → NVIC_PRI8_R [15:13]
UART3 → NVIC_PRI15_R [15:13]

PRIn Register Bit Field	Interrupt Source	Priority Register Macros
Bits 31:29	Interrupt[IRQ] = Interrupt[4n + 3]	NVIC_PRI**n**_R NVIC→IP[4**n**] – NVIC→IP[4**n** + 3]
Bits 23:21	Interrupt[IRQ] = Interrupt[4n + 2]	
Bits 15:13	Interrupt[IRQ] = Interrupt[4n + 1]	
Bits 7:5	Interrupt[IRQ] = Interrupt[4n]	

COMUNICACIÓN SERIAL - UART

INTERRUPCIONES

- REGISTRO DE PRIORIDAD Y HABILITACION DE INTERRUPCION (NVIC)
  - REGISTRO PARA HABILITAR LA INTERRUPCION (NVIC_ISERx)
    - NVIC_EN0_R: Provides 32 bit-by-bit enable control ability to 32 peripherals whose IRQ numbers are 0~31 (see Table 5.10 for IRQ numbers). This means that bit 0 controls the peripheral whose IRQ number is 0 (GPIO Port A), bit 1 controls the peripheral whose IRQ number is 1 (GPIO Port B), and bit 31 controls the peripheral whose IRQ number is 31 (GPIO Port G).
    - NVIC_EN1_R: Provides 32 bit-by-bit enable control ability to 32 peripherals whose IRQ numbers are 32~63.
    - NVIC_EN2_R: Provides 32 bit-by-bit enable control ability to 32 peripherals whose IRQ numbers are 64~95.
    - NVIC_EN3_R: Provides 32 bit-by-bit enable control ability to 32 peripherals whose IRQ numbers are 96~127.
    - NVIC_EN4_R: Provides 11 bit-by-bit enable control ability to 10 peripherals whose IRQ numbers are 128~138.

$$\textbf{Target bit number} = \textbf{IRQ number} - \textbf{32} \times (\textbf{n} - \textbf{1}).\ \textbf{n} = \textbf{1}\ \text{if}\ \text{IRQ} < 31,$$
$$\textbf{n} = \textbf{2}\ \text{if}\ (64 > \text{IRQ} > 31),$$
$$\textbf{n} = \textbf{3}\ \text{if}\ (96 > \text{IRQ} > 63),\ \textbf{n} = \textbf{4}\ \text{if}\ (128 > \text{IRQ} > 95).$$

UART0 → NVIC_EN0 [5]
UART1 → NVIC_EN0 [6]
UART2 → NVIC_EN1 [1]

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO